Linear Regression, Cost Functions

Jacob J. Sauerhoefer
www.jacobsauerhoefer.com
Dec. 8th 2024

Introduction

In linear regression, we form linear models to predict scalar-valued quantities, such as expected prices, based on data from some measurable inputs. Here, we outline some of the fundamental notation and terminology used across Machine Learning, Statistics and Econometrics when building linear models and performing linear regression. This article follows [1] and [2], providing additional explanations and visualizations as needed.

Linear Regression

Following the notation of [1], suppose we were interested in investigating:

$$Y \mapsto y = \text{the expected price of a home}$$

To determine a y that represents the real world as accurately as possible, we can identify a number of related *random variables* (X_1, X_2, X_3, etc.)[1]:

$$\begin{aligned} X_1 &\mapsto x_1 = \text{distance to city center} \\ X_2 &\mapsto x_2 = \text{square feet} \\ X_3 &\mapsto x_3 = \text{number of rooms} \\ &\dots \text{ etc.} \end{aligned}$$

[1]From an experimental perspective, we can assume that Y and X are *random variables* since real world observations are almost always subject to underlying variability or uncertainty [3].

Once we have established a goal and some observable factors, we can start to define a linear model. This linear model establishes a set of **hypotheses**, possible relationships between our measureable factors and output variable. These hypotheses can be written as a system of equations:

$$\begin{aligned} x_{11}w_1 + x_{12}w_2 + \dots + x_{1n}w_n &= y_1 \\ x_{21}w_1 + x_{22}w_2 + \dots + x_{2n}w_n &= y_2 \\ &\vdots \\ x_{m1}w_1 + x_{m2}w_2 + \dots + x_{mn}w_n &= y_m \end{aligned}$$

Or more compactly:

$$\sum_{n=1}^{N} x_{mn}w_n = y_m$$

Here, we've parameterized our model with **weights** w. As a matrix-vector, this system has the form:[2]

$$\mathbf{X}\mathbf{w} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix} = \mathbf{y}$$

[2]Instead of **Xw=y** and w, we could have used the notation β and $\mathbf{X}\beta$=**y**. Overalapping terminology in Linear Algebra, Statistics and Machine Learning is summarized in the appendix at the end.

The goal is to determine a vector $\mathbf{w} = (w_1, \dots, w_n)^T \in \mathbb{R}^n$ that solves the above equation.

Existence of exact solutions for Xw=y

The *column space* of a matrix can be thought of the as the span of the vectors that are formed by its columns. For the relationship **Xw**=**y**, there is an exact solution when $\mathbf{y}$ is in the column space of $\mathbf{X}$ with some combination of weights in $\mathbf{w}$.[3]

[3] Another way to say this is that an exact solution exists if y is in the subspace reached by stretching and scaling the columns of the design matrix X.

In other words, the observed data points $\mathbf{y}$ can be exactly expressed as the linear combination expressed earlier[4]:

[4] Equations of this form satisfy the definition of a **hyperplane**, a generalization of a plane-like object to an n-1 subspace in an n-dimensional ambient space.

$$x_{m1}w_1 + x_{m2}w_2 + \cdots + x_{mn}w_n = y_m$$

where x_1, x_2 are the columns (features) of $\mathbf{X}$ and there exist a $w_1, w_2, \ldots$ that result in $\mathbf{y}$. Lets consider a hypothetical data set with two features and a hyperplane to represent a fitted linear model:

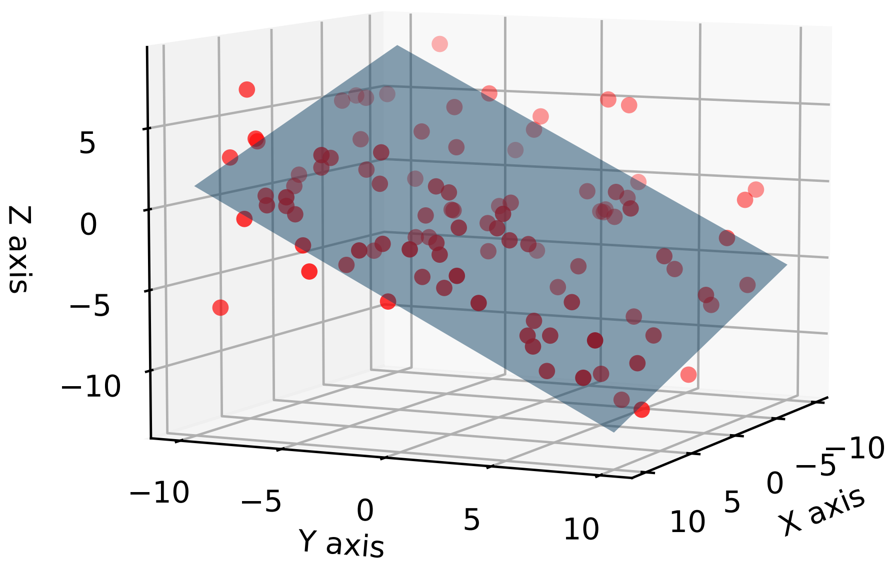

Figure 1: Hypothetical linear model in $\mathbb{R}^3$ with 2 features (Y and X axis) and target variable (Z axis).

As we can see, any amount of scatter in data prevents a linear model from being fitted exactly. We'll need to instead:

1. Find an approximate solution.
2. Quantify how good that approximation is.

The concept of **residuals** helps us do this.

Residuals, Loss & Cost Functions

Residuals, notatated with ε (also called **regression errors**) help us assess how far off an approximate solution is for a linear model. Since we are now estimating a solution, we'll change some notation and introduce **estimators** $\hat{\mathbf{y}}$ and $\hat{\mathbf{w}}$:

$$\hat{\mathbf{y}} = \mathbf{X}\hat{\mathbf{w}} = \mathbf{X}\mathbf{w} + \varepsilon$$
$$\hat{\mathbf{y}} = \mathbf{y} + \varepsilon$$
$$\varepsilon = \hat{\mathbf{y}} - \mathbf{y}$$

Here, the residuals ε quantify the difference between our estimated target vector $\hat{\mathbf{y}}$ and actual output vector $\mathbf{y}$.

Loss Functions

We'll extend the idea of residuals to define a **loss function**[5] in terms of individual data points, where we compare some true value y_i to some target $\hat{y}_i$:

$$L(y_i, \hat{y}_i)$$

[5] The terms *cost*, *loss* and *error* are often used interchangeably [4]. Here, we differentiate between the terms *loss* and *cost* to highlight the general approach to error evaluation.

Absolute Error (L1 Loss)	**Squared Error** (L2 Loss)
$L(y_i, \hat{y}_i) = \lvert y_i - \hat{y}_i \rvert$	$L(y_i, \hat{y}_i) = (y_i - \hat{y}_i)^2$

Cost Function

A loss function can be aggregated over multiple data points to form a **cost function**. Cost functions are usually evaluated to form a single number that represents the overall error for the model:

$$J(\mathbf{w})$$

Mean Absolute Error (MAE)	**Mean Squared Error** (MSE)
$J(y, \hat{y}) = \frac{1}{n}\sum(\lvert y - \hat{y} \rvert)$	$J(y, \hat{y}) = \frac{1}{n}\sum\left((y - \hat{y})^2\right)$

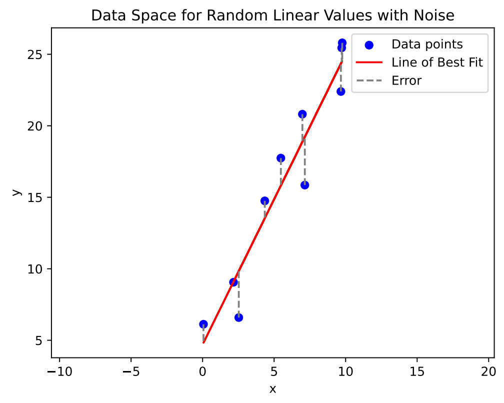

Figure 2: Absolute errors plotted as a difference from line of best fit.

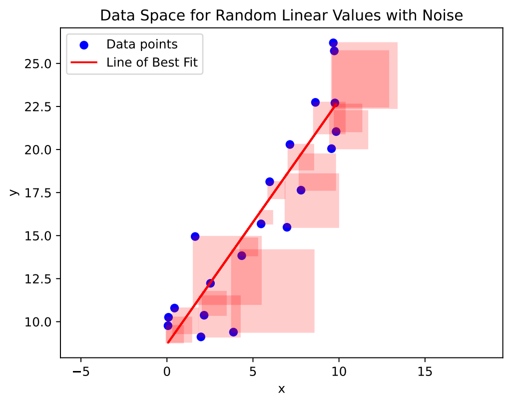

Figure 3: Squared errors plotted as a difference from line of best fit.

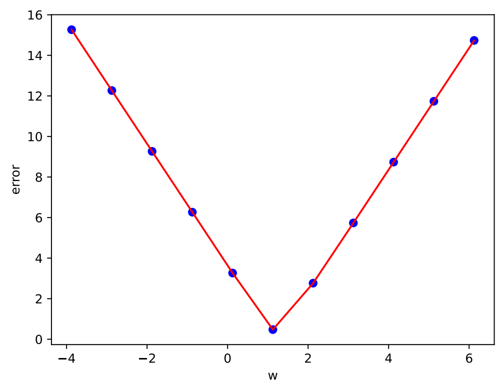

Figure 4: Absolute errors

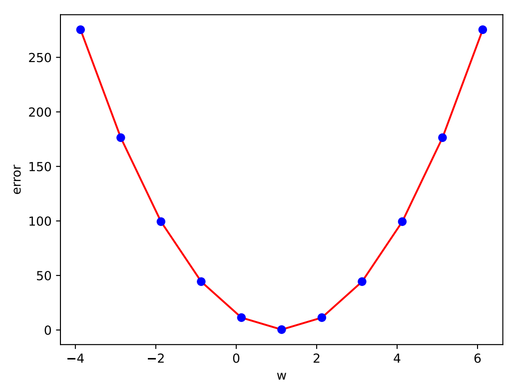

Figure 5: Squared errors

Comparing the Error Plot for the MAE and MSE, we see that the main characteristic of the MSE is its steeper curve. Errors farther from the minimum are more penalized when compared the the MAE.

Appendix

Matrix Vector Terminology

Linear systems and linear model fitting arises in multiple diciplines and each has its own distinct notation. Expanding on a table from [5], this matrix-vector is often notated in different ways:

Ours	LinAlg	Stats	Common Names
$\mathbf{X}\mathbf{w} = \mathbf{y}$	$\mathbf{A}\mathbf{x} = \mathbf{b}$	$\mathbf{X}\beta = \mathbf{y}$	Linear Model
$\mathbf{X}$	$\mathbf{A}$	$\mathbf{X}$	Feature / Design Matrix (columns = independent variable, rows= observations or hypotheses), predictors, training data, features
$\mathbf{w}$	$\mathbf{x}$	β	Regression coefficients or weights or beta parameters
$\mathbf{y}$	$\mathbf{b}$	$\mathbf{y}$	Dependent, Output or Target Variable, Realization Vector

Cases for Design Matrix $\mathbf{X}$

Type	Case	Characteristic	Possible Solutions
Overdetermined	m > n	tall and skinny	No solution, unique solution (by approximation), or none (rare)
Square	m = n	square	Infinitely many solutions or no solution
Underdetermined	m < n	wide and short	Unique solution (by approximation), no solution, or infinitely many solutions

Bibliography

[1] S. Hubbert, *Essential Mathematics for Market Risk Management*, 2nd ed. in Wiley Finance Series. Chichester, West Sussex, UK: Wiley, 2012.

[2] K. Krzyk, "Cost Function of Linear Regression: Deep Learning for Beginners." [Online]. Available: https://builtin.com/machine-learning/cost-function

[3] BigBendRegion, "What Is a Random Variable and What Isn't in Regression Models." Accessed: Nov. 24, 2024. [Online]. Available: https://stats.stackexchange.com/questions/485011/what-is-a-random-variable-and-what-isnt-in-regression-models

[4] StackExchange, "Objective Function, Cost Function, Loss Function: Are They the Same Thing?." Accessed: Nov. 27, 2024. [Online]. Available: https://stats.stackexchange.com/questions/179026/objective-function-cost-function-loss-function-are-they-the-same-thing

[5] J. Hefferon, *Linear Algebra*, Fourth. 2020.